

## Partie 3

# Interruptions



[khalil.hachicha@gmail.com](mailto:khalil.hachicha@gmail.com)

# Principe des Interruptions



- Une interruption est un événement interne ou externe au microcontrôleur qui va provoquer:
  - Une suspension du programme principal en cours d'exécution
  - L'exécution d'un sous programme dit d'interruption
    - Interrupt Service Routine (ISR)
  - La reprise – à l'endroit où il a été interrompu – de l'exécution du programme principal lorsque le sous programme d'interruption est terminé

# Intérêt des Interruptions

- Permet de lancer une tâche sur un périphérique (externe ou interne) et de récupérer le résultat sans avoir à attendre et bloquer le reste du programme
- Pas besoin de boucles de scrutation (polling) qui bloque le programme
- Permet de consacrer le temps d'exécution du processeur à d'autres tâches plus utiles

# Déroulement dans le $\mu$ P

- Sauvegarde des registres du  $\mu$ P dans la pile mémoire
  - Notamment le registre PC (Program Counter)
  - Sauvegarde de contexte
- Le PC va indiquer l'adresse de début du sous programme d'interruption
- Exécution du sous programme d'interruption
- Restauration des registres
  - Récupération de contexte
  - Le registre PC indique l'adresse de l'instruction où s'est arrêté le programme principal
- Reprise du programme principal

# Interruptions et LPC2378

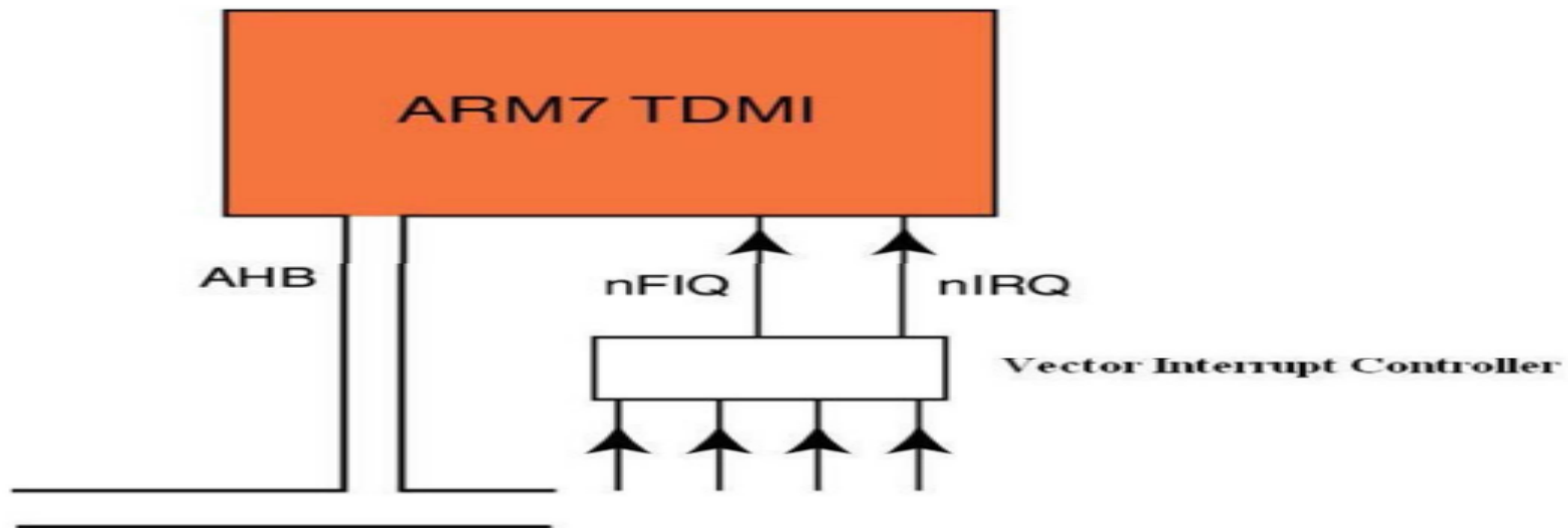
- L'ARM7 possède deux lignes d'interruption
  - Mode Fast Interrupt Request (FIQ)
  - Mode Interrupt Request (IRQ)
- Le LPC2378 propose 32 sources d'interruption.

N° Canal	Source
0	Watchdog Timer
1	Reserved for software interrupts only
2	ARM Core Embedded ICE, DbgCommRx
3	ARM Core Embedded ICE, DbgCommTx
4	TIMER0
5	TIMER1
6	UART0
7	UART1
8	PWM1
9	I2C0
10	SPI0, SSP0
11	SSP1
12	PLL
13	RTC
14	External Interrupt 0
15	External Interrupt 1

16	External Interrupt 2
17	External Interrupt 3
18	A/D Converter
19	I2C1
20	Brown Out Detect
21	Ethernet
22	USB
23	CAN
24	SD / MMC
25	GP DMA
26	Timer 2
27	Timer 3
28	UART 2
29	UART 3
30	I2C2
31	I2S

# Contrôleur d'interruption

- Comment lier les 32 sources du LPC2378 aux 2 lignes de l'ARM7?
- A l'aide d'un contrôleur d'interruption
  - Contrôleur d'Interruption Vectorisé (VIC)
  - 1 canal associé à chaque source (voir Table)



# Activation des Interruptions

- Registre VICIntEnable
  - Autorise les sources d'interruptions
  - Registre 32 bits (1 bit par source)
  - Le bit i est associé à la source de canal i

Bit	Symbol	Description	Reset value
31:0	See <a href="#">Table 7-78 "Interrupt sources bit allocation table"</a> .	When this register is read, 1s indicate interrupt requests or software interrupts that are enabled to contribute to FIQ or IRQ. When this register is written, ones enable interrupt requests or software interrupts to contribute to FIQ or IRQ, zeroes have no effect. See <a href="#">Section 7-4.5 "Interrupt Enable Clear Register (VICIntEnClear - 0xFFFF F014)" on page 80</a> and <a href="#">Table 7-68</a> below for how to disable interrupts.	0

- Les sources d'interruptions qui ne sont pas autorisées sont dites "masquées"
  - Au reset, toutes les sources sont masquées
  - Le programmeur autorise les sources qu'il va utiliser dans son application

# Annulation d'une Interruption

- Ecrire un '0' dans VICIntEnable n'a aucun effet
- Pour désactiver une interruption, il faut modifier le registre VICIntEnClear

Bit	Symbol	Value	Description	Reset value
31:0	<a href="#">See Table 7-78 "Interrupt sources bit allocation table".</a>	0	Writing a 0 leaves the corresponding bit in VICIntEnable unchanged.	-
		1	Writing a 1 clears the corresponding bit in the Interrupt Enable register, thus disabling interrupts for this request.	

- La mise à '1' d'un bit de VICIntEnClear met également à '0' le bit correspondant dans VICIntEnable



# Sélection du Mode d'Interruption

- Le registre VICIntSelect permet de choisir le mode (FIQ ou IRQ) de l'interruption.

Table 69. Interrupt Select register (VICIntSelect - address 0xFFFF F00C) bit description

Bit	Symbol	Value	Description	Reset value
31:0	<a href="#">See Table 7-78 "Interrupt sources bit allocation table".</a>	0	The interrupt request with this bit number is assigned to the IRQ category.	0
		1	The interrupt request with this bit number is assigned to the FIQ category.	

- Dans ce cours, nous traitons uniquement le mode IRQ.

# Interruptions Vectorisées

- Dans la plupart des processeurs actuels, les interruptions sont vectorisées
  - Dans l'ARM7, c'est le cas des interruptions en mode IRQ
- A chaque source d'interruption IRQ est associée une adresse dans la mémoire programme :
  - Cette adresse s'appelle le vecteur d'interruption
  - Elle correspond à la première instruction du sous programme d'interruption associée à la source d'interruption
  - Ce mécanisme diminue le temps de réponse aux interruptions

# Interruptions Vectorisées en C

- Le compilateur KEIL permet de définir les ISR vectorisées en C
  - Grâce à une extension au langage C ANSI spécifique à l'outil KEIL
  - En-tête du sous-programme spécifique avec des mots-clés obligatoires

```
void my_isr(void) _ _ irq
```

# Adresse des ISR Vectorisées

- Chaque canal du VIC possède un registre d'adresse
  - VICVectAddr0 → VICVectAddr31
  - VICVectAddrN associé au canal N
- On y écrit l'adresse de début de l'ISR associée à la source d'interruption
- Exemple en C (KEIL)

```
VICVectAddrN= (unsigned long) my_isr;
```

# Priorité des IRQ Vectorisées

- Lorsque plusieurs sources d'interruption sont autorisées, il faut définir une priorité au cas où il y ait plusieurs requêtes simultanées
- Par défaut, les interruptions avec le plus petit numéro de canal sont prioritaires
- Les registres VICVectPriorityN permettent de définir un niveau de priorité pour chaque source

Table 73. Vector Priority registers 0-31 (VICVectPriority0-31 - addresses 0xFFFF F200 to 0xFFFF F27C) bit description

Bit	Symbol	Description	Reset value
3:0	VICVectPriority	Selects one of 16 priority levels for the corresponding vectored interrupt.	0xF
31:4	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

- 0: Priorité la plus forte                      15: Priorité la plus faible

# Préemption des IRQ Vectorisées



- Si un sous-programme d'interruption est en cours d'exécution, il ne peut pas être interrompu(\*)
- On dit que les interruptions de l'ARM7 ne sont pas préemptables
- (\*) Il existe un mécanisme (fonction logicielle) qui permet néanmoins de le faire.

# Sous Programme d'Interruption



- Une ISR n'a pas de paramètres d'entrée ni de valeur de retour
- Pour échanger des informations entre une ISR et un autre programme, il faut utiliser des variables globales

# Sous programme d'interruption

- A la fin de l'ISR, il faut indiquer que la requête d'interruption a été traitée
- Cela se fait en deux temps
  - Il faut remettre à 0 le drapeau de demande d'interruption
    - Ce drapeau se situe dans un registre du périphérique qui a fait la demande d'IRQ.
  - Il faut remettre à 0 le Registre VICVectAddr du VIC.
    - C'est dans ce registre que le VIC a copié le vecteur de l'interruption qui a été traitée.

**VICVectAddr** = 0x00000000 ;